Army Research Laboratory

Automated Target Recognizer for the Demo III Program

Sandor Der

ARL-TR-1569 July 2002

Approved for public release; distribution unlimited.

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-TR-1569 **July 2002**

Automated Target Recognizer for the Demo III Program

Sandor Der
Sensors and Electron Devices Directorate

Abstract

This report describes an algorithm for the recognition of military vehicles in Forward Looking Infrared (FLIR) imagery. The input is a FLIR image, and the output of a detector or clutter rejector listing a number of locations in the image for the recognizer to examine. The output is the same list with the decision of the recognizer appended to each location in the list. The algorithm is based on principal component analysis.

Contents

Figures

Table

1. Introduction

This work was performed by the Demo III Unmanned Ground Vehicle (UGV) program, which is developing UGVs that will assist Army scouts. The Electro-Optics Infrared (EOIR) Image Processing branch (AMSRL-SE-SE) has been tasked with developing algorithms for acquiring and recognizing targets imaged by the Wescam Forward Looking Infrared (FLIR) sensor. These images are sent back to the user upon request, or when the automatic target recognizer (ATR) indicates a location of interest. The user makes the ultimate decision about whether an object in an image is actually a target. The ATR reduces the bandwidth requirement of the communication link, because the imagery can be sent back at reduced resolution, except those regions indicated by the ATR as being possible targets. The algorithms consist of a front-end detector, a clutter rejector, and a recognizer. This paper describes only the recognizer.

The algorithm described in this report was designed to address a need for a recognition algorithm that could be trained with a small amount of data, with poor range and localization information. The operational scenario examines objects that have been detected by another algorithm to determine if they are one of the objects stored in an existing image library. The detection location given by the detection algorithm may be poorly centered on the target, and the range to the target will not be known. The number of training examples for the four different targets differed radically. This meant that the chosen algorithm must be able to take advantage of a large training set when it exists, but still be able to perform well for smaller data sets.

Many techniques have been applied to the recognition problem [1]. When training sets have been large, recognition algorithms have typically used complex learning algorithms that use a large number of features to discriminate between targets. Often the features are either simply the pixel values, or simple gradient/wavelet features calculated in a dense grid across the target region [3–5]. The learning algorithms include complex template matching schemes [5] or neural networks [2–4,6,7]. Learning algorithms that are trained on small data sets tend to generalize poorly, so we chose not to use these algorithms for this work.

Some algorithm designers have used principal component analysis (PCA) to compress the data prior to recognition [8]. An advantage of this approach is that it reduces the number of features that a classifier can use, and thus reduces the size of the required training set. One disadvantage is that the compression eliminates some of the information that is useful to perform discrimination, and since the PCA algorithm optimizes for compression of the data without regard for information that is useful for discrimination, one cannot expect that PCA gives the most discrimination information possible for a given number of features.

The data set used for our training was lopsided. The algorithm attempts to recognize four targets, two real (M113 and HMMWV) and two target boards (TB1 and TB2). For the M113 and HMMWV, we have 1239 and 2080 suitable training samples, while for the target boards we have 14 and 22 suitable training samples. The ramifications of this imbalance will be discussed.

The remainder of this technote is organized as follows. Section 2 describes the data used to train and test the system. Section 3 describes the architecture of the recognizer. Section 4 gives the results of experiments performed on a small test set of imagery. Section 5 contains conclusions and plans for future work.

2. The Data

The training and testing data were gathered from various sources. The testing set consisted of suitable images from the Fort Indiantown Gap data collection of 2001. Images were selected that met a number of conditions. The images must contain the target at sufficient resolution for human recognition. The target images must be nearly unoccluded. Fort Indiantown Gap data was chosen for testing since the actual demo scheduled for September 2001 is to be located there. Also, this data was taken with the same sensor configuration that will be used for the demo. The test data contained 64 images of the M113, and 45 images of the target boards.

The training set consisted of images taken with previous configurations of the sensor, or with another sensor. Because the amount of data from the latest sensor configuration was small, we decided to save all of it for testing, and obtain data from other sources for training. The test on the most appropriate data would let us know if the outside data was unsuitable. The only other data of the target boards was obtained with the same sensor, but with an eight bit digitizer. This resulted in many saturated images; in particular, target regions were likely to be saturated. Other data of the M113 and HMMWV included data from previous versions of the sensor, and from different sensors that we had on hand.

For the training data, we had 14 and 22 images of the target boards, and 1239 and 2080 images of the M113 and HMMWV.

The lopsided training set suggests an algorithm architecture that can handle a wide variation of training set size and variability. The numbers given above overstate the problem for a couple of reasons. The target boards are two dimensional plywood boards with attached heating panels, and as such there is essentially one pose. The real vehicles can be seen from arbitrary azimuth angle, and some variation in elevation. The algorithm groups all of these poses into four groups for the purpose of PCA eigenvector generation. Also, the signature of the target boards is not nearly as variable as the real targets. The target boards have fixed heating panels, so the greatest variability is the angle of view variation, and the relative temperatures of the heating panels, the bare plywood, and the background. The solar irradiation on the panel should be nearly constant, since the panel is flat. The real target signatures vary because of the amount of solar irradiance differs on different portions of the target, the exercise state effects different parts differently (hot wheels if there has been movement, hot engine if engine is running, regardless of movement, etc.), and the pose varies. Still, it can be expected that the training set of the target boards does not capture the variability as well as for the real vehicles, and it is therefore important that a bias reduction technique is used after the PCA transformation.

3. Algorithm Architecture

We have chosen a PCA decomposition/reconstruction technique for the algorithm. The idea is to calculate a PCA decomposition of each target-pose group using the training set. For testing, each target is decomposed using the first n PCA eigenvectors, then reconstructed, and the mean square error (MSE) of the difference between the original and reconstructed target is calculated. This gives one MSE value for each target-pose group. The minimum reconstruction error should occur for the correct target-pose group. Because the PCA captures a different proportion of the total information for each of the target-pose groups, the MSE values are adjusted by a weighting vector prior to choosing the minimum value. We emphasize that the data set drives the choice of algorithm architecture.

3.1 Introduction to Principal Component Analysis

The PCA decomposition does not capture all of the information in the input target, because the decomposition is truncated at some small number n of eigenvectors. Also referred to as the Hotelling transform or the discrete Karhunen-Loève transform, PCA is based on statistical properties of vector representations. PCA is an important tool for image processing because it has several useful properties, such as decorrelation of data and compaction of information (energy). We provide here a summary of the basic theory of PCA.

Assume a population of random vectors of the form

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_3 \end{bmatrix}. \tag{1}$$

The *mean vector* and the *covariance matrix* of the vector population **x** are defined as

$$\mathbf{m_x} = E\{x\}, \text{ and} \tag{2}$$

$$\mathbf{C_x} = E\left\{(\mathbf{x} - \mathbf{m_x})(\mathbf{x} - \mathbf{m_x})^{\mathbf{T}}\right\}, \tag{3}$$

where $E\{\text{arg}\}$ is the expected value of the argument, and T indicates vector transposition. Because $\mathbf{x}$ is n-dimensional, $\mathbf{C_x}$ is a matrix of order $n \times n$. Element c_{ii} of $\mathbf{C_x}$ is the variance of x_i (the ith component of the $\mathbf{x}$ vectors in the population), and element c_{ij} of $\mathbf{C_x}$ is the covariance between elements x_i and x_j of these vectors. The matrix $\mathbf{C_x}$ is real and symmetric. If elements x_i and x_j are uncorrelated, their covariance is zero and, therefore, $c_{ij} = c_{ji} = 0$. For N vector samples from a random population, the mean vector and covariance matrix can be approximated from the samples by

$$\mathbf{m_x} = \frac{1}{N}\sum_{p=1}^{N}\mathbf{x_p}\,, \text{ and} \tag{4}$$

$$\mathbf{C_x} = \frac{1}{N}\sum_{p=1}^{N}\left(\mathbf{x_p}\mathbf{x_p^T} - \mathbf{m_x}\mathbf{m_x^T}\right) \tag{5}$$

Because $\mathbf{C_x}$ is real and symmetric, we can always find a set of n orthonormal eigenvectors for this covariance matrix.

Let $\mathbf{e_i}$ and λ_i, $i = 1, 2, \ldots, n$, be the eigenvectors and the corresponding eigenvalues of $\mathbf{C_x}$, sorted in a descending order so that $\lambda_j \geq \lambda_{j+1}$ for $j = 1, 2, \ldots, n-1$. Let $\mathbf{A}$ be a matrix whose rows are formed from the eigenvectors of $\mathbf{C_x}$, such that

$$\mathbf{A} = \begin{bmatrix} \mathbf{e_1} \\ \mathbf{e_2} \\ \vdots \\ \mathbf{e_n} \end{bmatrix}. \tag{6}$$

This $\mathbf{A}$ matrix can be used as a linear transformation matrix that maps the $\mathbf{x}$'s into vectors, denoted by $\mathbf{y}$'s, as follows:

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m_x}) \tag{7}$$

Conversely, we may want to reconstruct vector $\mathbf{x}$ from vector $\mathbf{y}$. Because the rows of $\mathbf{A}$ are orthonormal vectors, $\mathbf{A}^{-1} = \mathbf{A}^{\mathrm{T}}$. Therefore, any vector $\mathbf{x}$ can be reconstructed from its corresponding $\mathbf{y}$ by the relation

$$\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{y} + \mathbf{m_x}\,. \tag{8}$$

Instead of using all the eigenvectors of $\mathbf{C_x}$, we may pick only k eigenvectors corresponding to the k largest eigenvalues and form a new transformation matrix $\mathbf{A}_k$ of order $k \times n$. In this case, the resulting $\mathbf{y}$ vectors would be k-dimensional, and the reconstruction given in equation (8) would no longer be exact. The reconstructed vector using $\mathbf{A}_k$ is

$$\hat{\mathbf{x}} = \mathbf{A_k^T}\mathbf{y} + \mathbf{m_x} \tag{9}$$

The mean square error (MSE) between $\mathbf{x}$ and $\hat{\mathbf{x}}$ can be computed by the expression

$$\varepsilon = \sum_{j=1}^{n}\lambda_j - \sum_{j=1}^{k}\lambda_j = \sum_{j=k+1}^{n}\lambda_j\,. \tag{10}$$

Because the λ_j's decrease monotonically, equation (10) shows that we can minimize the error by selecting the k eigenvectors associated with the k largest eigenvalues. Thus, the PCA transform is optimal in the sense that it minimizes the MSE between the vectors $\mathbf{x}$ and their approximations $\hat{\mathbf{x}}$.

3.2 PCA Decomposition/Reconstruction Architecture

The PCA decomposition described above takes a training set of images and turns them into an ordered set of eigenvectors and corresponding eigenvalues. This decomposition is performed for each target-pose group of training samples. Since there are two real targets, which are divided into four pose groups each, and two target board types, which only have one pose, there are a total of 10 target-pose groups. We have chosen the number of eigenvectors to retain to be 5, for a total of 50 stored eigenvectors. The eigenvectors are stored at different scales, as will be described later.

The decomposition stage determines the PCA components of a sample being tested. The components γ_i for an input target image $\mathbf{x}$ are calculated as

$$\gamma_i = \sum_{j=1}^{n} \mathbf{e}_{\mathbf{i},\mathbf{j}} \mathbf{x}_{\mathbf{j}} = \mathbf{e}_{\mathbf{i}} \cdot \mathbf{x} . \tag{11}$$

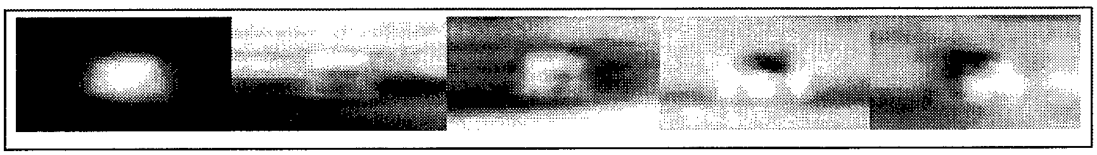

Figure 1. Eigenvectors of HMMWV front side.

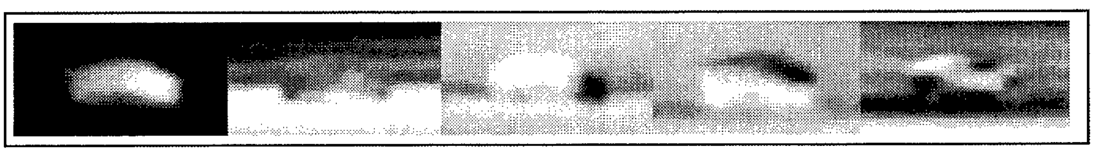

Figure 2. Eigenvectors of HMMWV left side.

Figure 3. Eigenvectors of HMMWV back side.

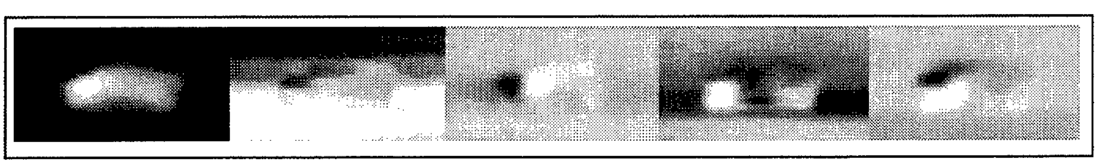

Figure 4. Eigenvectors of HMMWV right side.

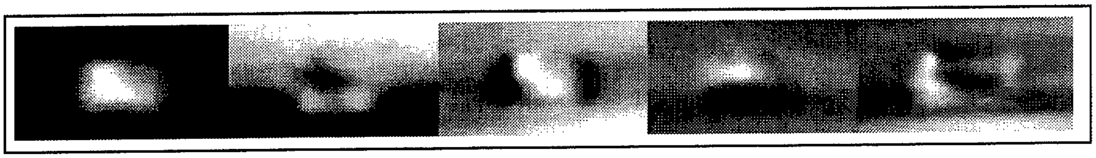

Figure 5. Eigenvectors of M113 front side.

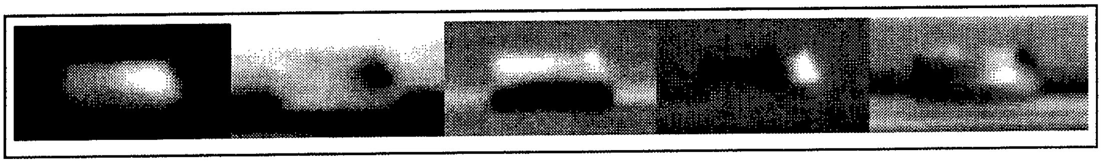

Figure 6. Eigenvectors of M113 left side.

Figure 7. Eigenvectors of M113 back side.

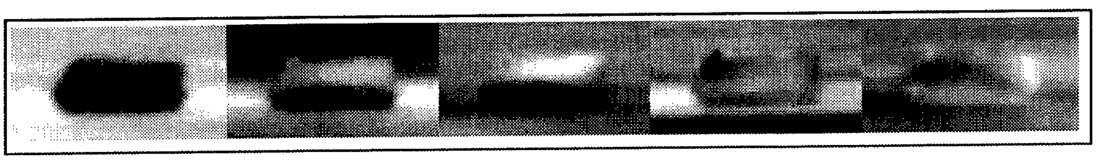

Figure 8. Eigenvectors of M113 right side.

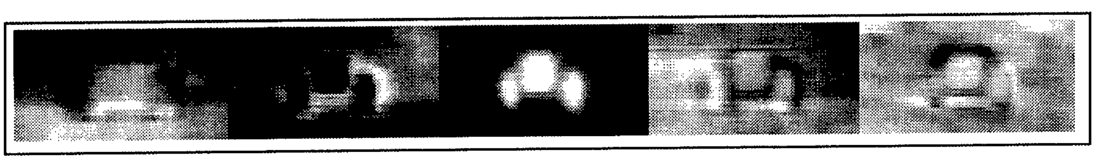

Figure 9. Eigenvectors of Target Board 1.

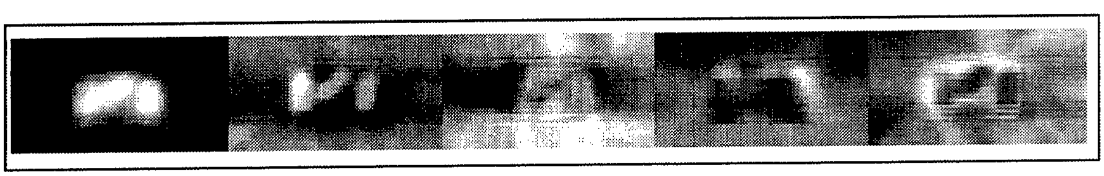

Figure 10. Eigenvectors of Target Board 2.

Thus the ith PCA component of an input vector $\mathbf{x}$ is simply the dot product of $\mathbf{x}$ with the ith eigenvector. The reconstruction using the first k eigenvectors is

$$\hat{x} = \sum_{i=1}^{k} \gamma_i \mathbf{e}_i \,. \tag{12}$$

The reconstruction error is simply

$$\varepsilon = \sum_{j=1}^{n} \left(\mathbf{x}_j - \hat{\mathbf{x}}_j\right)^2 = \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|^2 . \tag{13}$$

3.3 Linear Weighting of Reconstruction Error

To reduce the bias inherent in the PCA decomposition process, the reconstruction errors for each target-pose group are multiplied by a fixed weight. Thus the reconstruction error for the lth target-pose group, ε_l is weighted by a weight ω_l. The target-pose decision $\hat{l}$ is given by

$$\hat{l} = argmin_l\left(\omega_l \varepsilon_l\right) . \tag{14}$$

3.4 Scale and Shift Search Space

It is anticipated that this recognizer will be used after an automated detector has found potential targets in an image. It must be assumed that any detector will be imprecise about centering the detection on the target. For the Demo III application, the range to the target is also unknown, at least for some of the scenarios. Any template matching algorithm is inherently sensitive to translation and scale of the image. The algorithm was written to allow the user to specify the range uncertainty, as well as the translation uncertainty. If accurate range or translation is known, these will help algorithm performance. However, inaccurate information will degrade performance more than lack of information.

The algorithm handles this uncertainty by performing the decomposition/reconstruction operation at a number of different scales, and at a few location around the pixel indicated by the detector. Iterating through possible ranges and target locations increases the probability that a false target-pose will give a minimum reconstruction error.

The translation uncertainty is specified in pixels. The user is required to specify a minimum and maximum range; if nothing is known about the range to a target, the minimum and maximum ranges can be derived from knowledge of the sensor, and knowledge of the minimum resolution required by a recognizer. Range information can be derived from digital maps, shape from motion algorithms, or laser ranging. It is anticipated that for the current implementation, digital maps will be the only regular source of range information.

4. Experimental Results

A detection algorithm described elsewhere [9,10] was applied to the test imagery. The recognition algorithm takes as input the original image and the detection file produced by the detector. The recognizer was not given the ground truth center of the targets, only the detector estimated center. Table 1 shows the confusion matrix on the four class problem. The overall probability of correct identification is 59.63 percent.

Table 1. Confusion matrix on test set.

	HMMWV	M113	TB1	TB2
M113	6	41	30	14
TB1	1	11	10	1
TB2	0	6	2	14

Figure 11 shows a sample image that does not contain a target. Figures 12, 13, 14, and 15 contain targets. Some of the targets would be difficult for a human to distinguish. The target in figure 12 is difficult to distinguish because the shape is not clearly that of an M113, and there is little interior information because the whole target is hot. Figure 13 is clearly an M113, because of the rectangular plate on the upper front of the target is a distinguishing characteristic. Notice the target is not level; this makes recognition more difficult because the templates aren't well aligned. The algorithm doesn't currently tilt the templates to handle such a case. Doing so would make it more likely to correctly identify tilted targets, but would increase the probability of error on level targets, and would increase computation time. Figure 14 is a good example of a target that is difficult for an algorithm to detect, but easy for a human. The target does not have a clear boundary, nor is it hotter than its background. The detector and recognizer give correct results for this image, but that is unusual. Figure 15 shows a target board type II clearly visible in the left center of the image. While this is clearly a target board, it is not easy to see which type at this resolution.

5. Conclusions

We have presented a recognizer algorithm design for the demo3 program, and shown results on a small set of data from the demo3 sensor. The choice of architecture was driven by the size of the data set. Future work for the program will probably include adding a color tv camera to the algorithm, which might aid clutter rejection in the daytime.

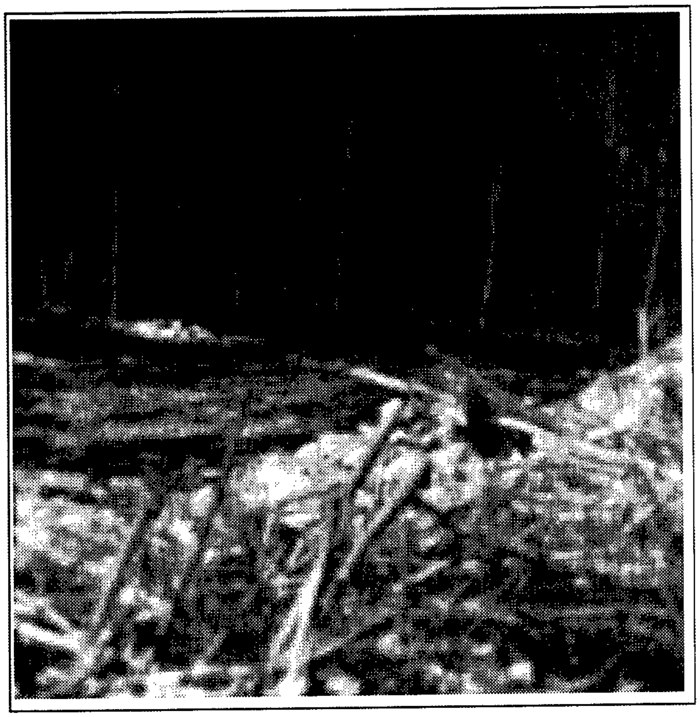

Figure 11. A Sample image, containing only clutter.

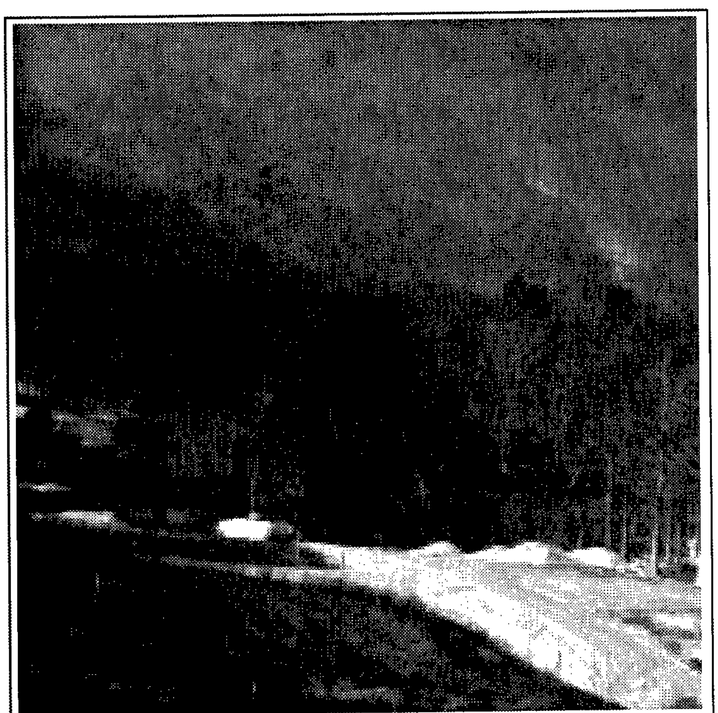

Figure 12. An image of the left side of an M113.

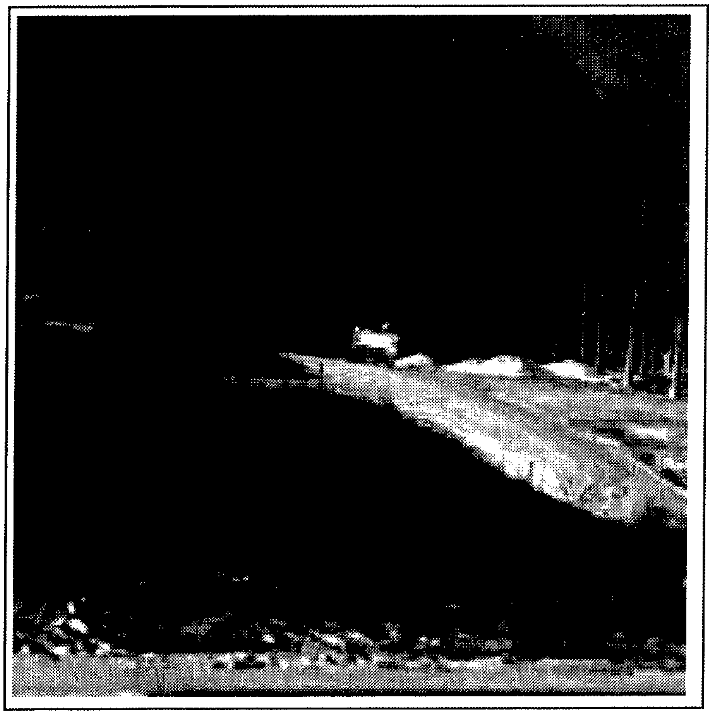

Figure 13. Front view of an M113.

Figure 14. Front view of an M113, on the road near the center of the image.

Figure 15. View of target board type II.

References

1. Bhanu, B., "Automatic target recognition: state of the art survey," *IEEE Trans. Aerospace Elect. Sys.*, 22(4), 364–379, (1986).

2. Roth, M. W., "Survey of Neural Network Technology for Automatic Target Recognition," *IEEE Trans. Neural Networks*, 1(1), 28–43, (1990).

3. Hecht-Nielsen R. and Y.-T. Zhou, "VARTAC: A Foveal Active Vision ATR System," *Neural Networks*, 8(7), 1309–1321, (1995).

4. Wang, L., S. Der, and N. Nasrabadi, "Modular Neural Network Recognition of Targets in FLIR Imagery," *IEEE Transactions on Image Processing*, Vol. 7, No 8, August 1998.

5. Chan, A. and Nasrabadi, N. (1997): Wavelet based vector quantization for automatic target recognition. International Journal on Artificial Intelligence Tools 6(2), 165–178.

6. Neubauer, C. (1998): Evaluation of convolutional Neural Networks for Visual Recognition. IEEE Transactions Neural Networks 9(4), 685–696.

7. Chen, C. H. and G. G. Lee G. G. (1996): Multi-resolution Wavelet Analysis Based Feature Extraction for Neural Network Classification. Proceedings International Conference Neural Networks 3, 1416–1421.

8. Chan, L., S. Der, and N. Nasrabadi, Analysis of Dualband FLIR Imagery for Automatic Target Detection, in Smart Imaging Systems, Bahram Javidi, ed., SPIE Press, 2001.

9. Der, S., C. Dwan, A. Chan, H. Kwon, N. Nasrabadi, "Scale Insensitive Vehicle Detection in Infrared Imagery," ARL Technical Report, 2000.

10. Kwon, H., S. Der, and N. Nasrabadi, "Multisensor target detection using adaptive feature-based fusion," SPIE Aerosense, April, 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 2002	Final, September 2000 to June 2001

4. TITLE AND SUBTITLE
Automated Target Recognizer for the Demo III Program

5. FUNDING NUMBERS
DA PR: AH16
PE: 62120A

6. AUTHOR(S)
Sandor Der

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL- SE-SE email: sder@arl.army.mil
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1569

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 2NE4M1
AMS code: 622120.H1600

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This report describes an algorithm for the recognition of military vehicles in Forward Looking Infrared (FLIR) imagery. The input is a FLIR image, and the output of a detector or clutter rejector listing a number of locations in the image for the recognizer to examine. The output is the same list with the decision of the recognizer appended to each location in the list. The algorithm is based on principal component analysis.

14. SUBJECT TERMS
target recognition, ATR, FLIR

15. NUMBER OF PAGES
19

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102